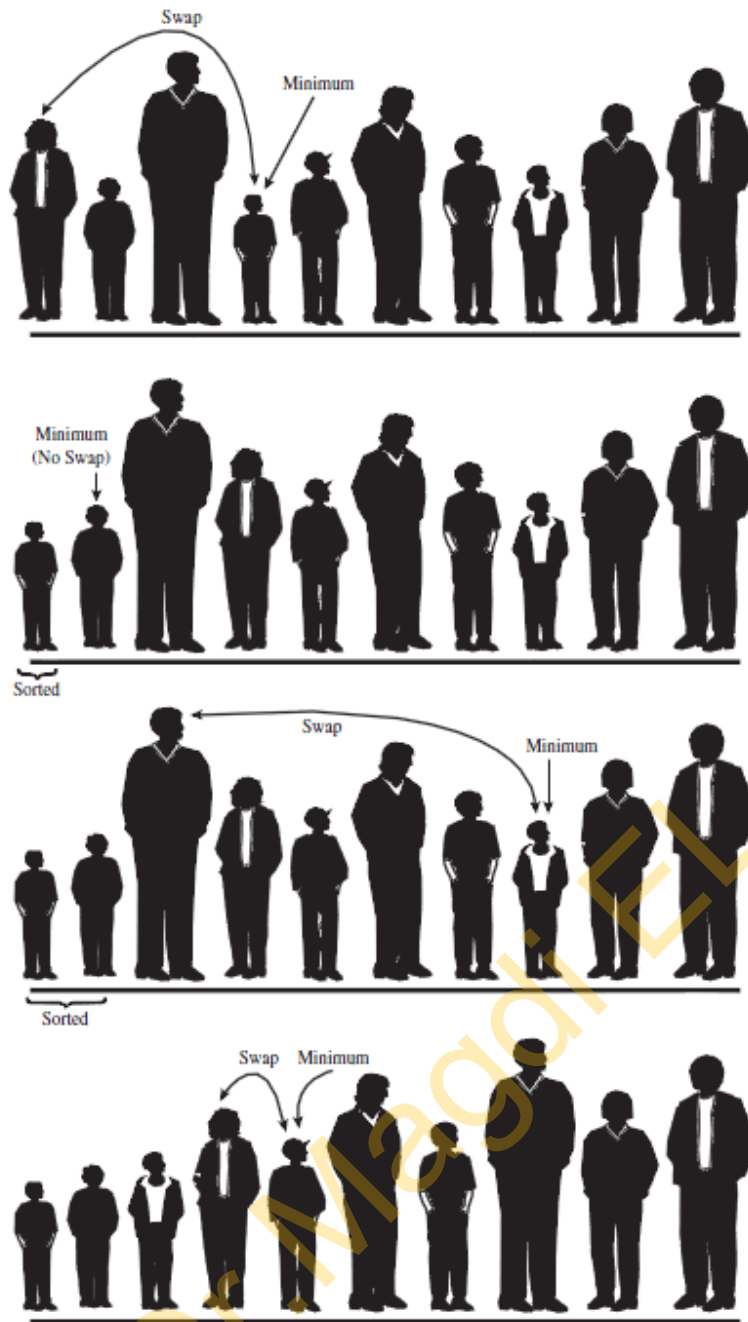


2- Selection sort



A Brief Description

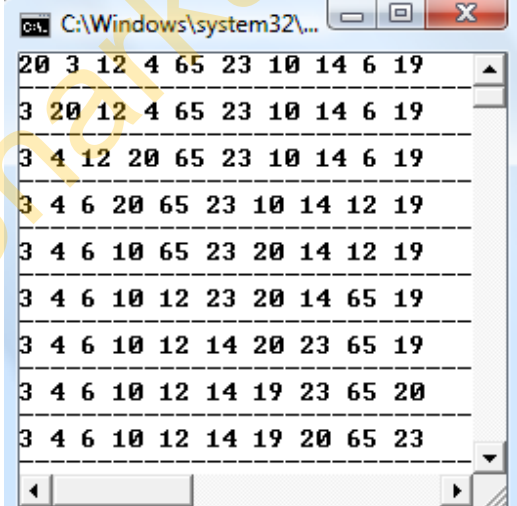
What's involved in the selection sort is making a pass through all the players and picking (or *selecting*, hence the name of the sort) the shortest one. This shortest player is then swapped with the player on the left end of the line, at position 0. Now the leftmost player is sorted and won't need to be moved again. Notice that in this algorithm the sorted players accumulate on the left (lower indices), whereas in the bubble sort they accumulated on the right.

The next time you pass down the row of players, you start at position 1, and, finding the minimum, swap with position 1. This process continues until all the players are sorted.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
int array1[10]={20,3,12,4,65,23,10,14,6,19};
int size=10;
void display();
int minimum(int);
void swap(int & , int &);
void main()
{
    display();
    for(int i=0; i < size-1; i++)
    {
        swap(array1[i], array1[minimum(i)]);
        display();
    }
}

int minimum(int pos)
{
    int min_val=array1[pos];
    int min_pos=pos;
    for(int k=pos+1; k<size; k++)
    {
        if(array1[k] < min_val)
        {
            min_val=array1[k];
            min_pos=k;
        }
    }
    return min_pos;
}

void swap (int &x, int &y)
{
    int temp=x;
    x=y;
    y=temp;
}
```



20	3	12	4	65	23	10	14	6	19
3	20	12	4	65	23	10	14	6	19
3	4	12	20	65	23	10	14	6	19
3	4	6	20	65	23	10	14	12	19
3	4	6	10	65	23	20	14	12	19
3	4	6	10	12	23	20	14	65	19
3	4	6	10	12	14	20	23	65	19
3	4	6	10	12	14	19	23	65	20
3	4	6	10	12	14	19	20	65	23

```
void display()
{
    for(int i=0;i<10;i++)
        cout << array1[i] << " ";
    cout << "\n" << "-----\n";
}
```

Efficiency of the Selection Sort

The selection sort performs the same number of comparisons as the bubble sort: $N*(N-1)/2$. For 10 data items, this is 45 comparisons. However, 10 items require fewer than 10 swaps. With 100 items, 4,950 comparisons are required, but fewer than 100 swaps. For large values of N , the comparison times will dominate, so we would have to say that the selection sort runs in $O(N^2)$ time, just as the bubble sort did. However, it is unquestionably faster because there are so few swaps. For smaller values of N , the selection sort may in fact be considerably faster, especially if the swap times are much larger than the comparison times.